REUSABLE PROBABILISTIC MODELS FOR SCIENTIFIC DATA

Michael Turmon
19 May 2000

A. Introduction and Motivation

B. Language

C. Initial Applications

D. Spatiotemporal Extensions

Joint work with Vlad Gluzman and Eric Mjolsness
of the JPL Machine Learning Group, and
Lukman Ramsey of JPL and UCSD
Institute for Neural Computation

`turmon@aig.jpl.nasa.gov`
`http://www-aig.jpl.nasa.gov/home/turmon/`

APPLICATION NEEDS

Goal

Allow scientists to define and exchange statistical models for data

- Model definition
- Model interchange
- Container to facilitate computation
- Backed by computational engine(s)

Applications

Scientific problems in which observable variables relate to hidden variables:

Remote sensing (find solar features: observables → labels)
Clustering (gene expression array analysis: next page)
Time series (HMMs for environmental time series)

Specifics

Ability to handle continuous variables

Support real-valued transformations

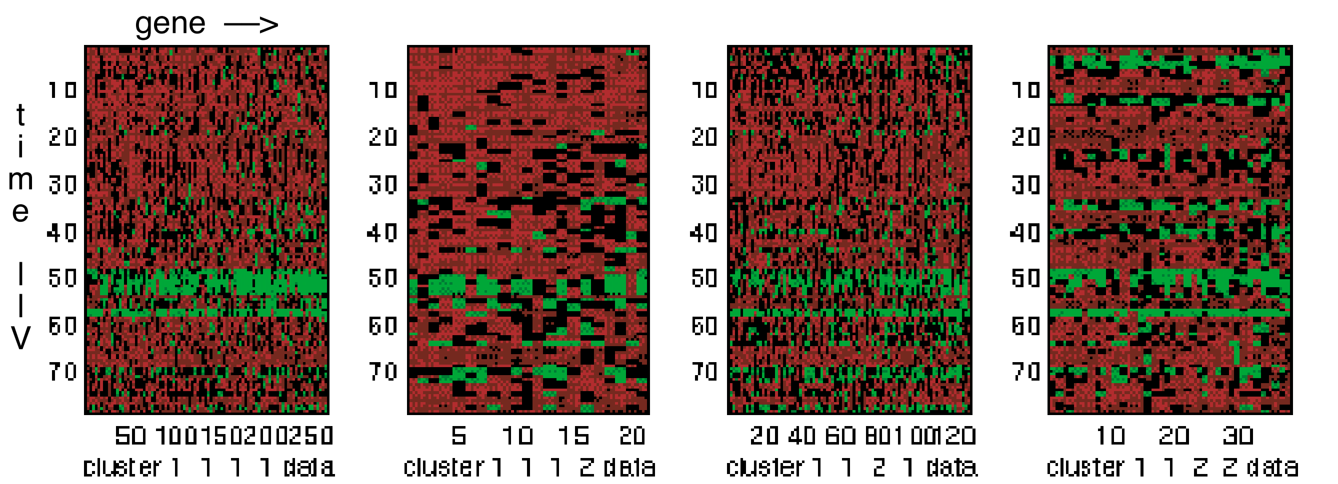
Restricted family of models is enough at first

Mostly bottom-up inference: not complex diagnostic setting

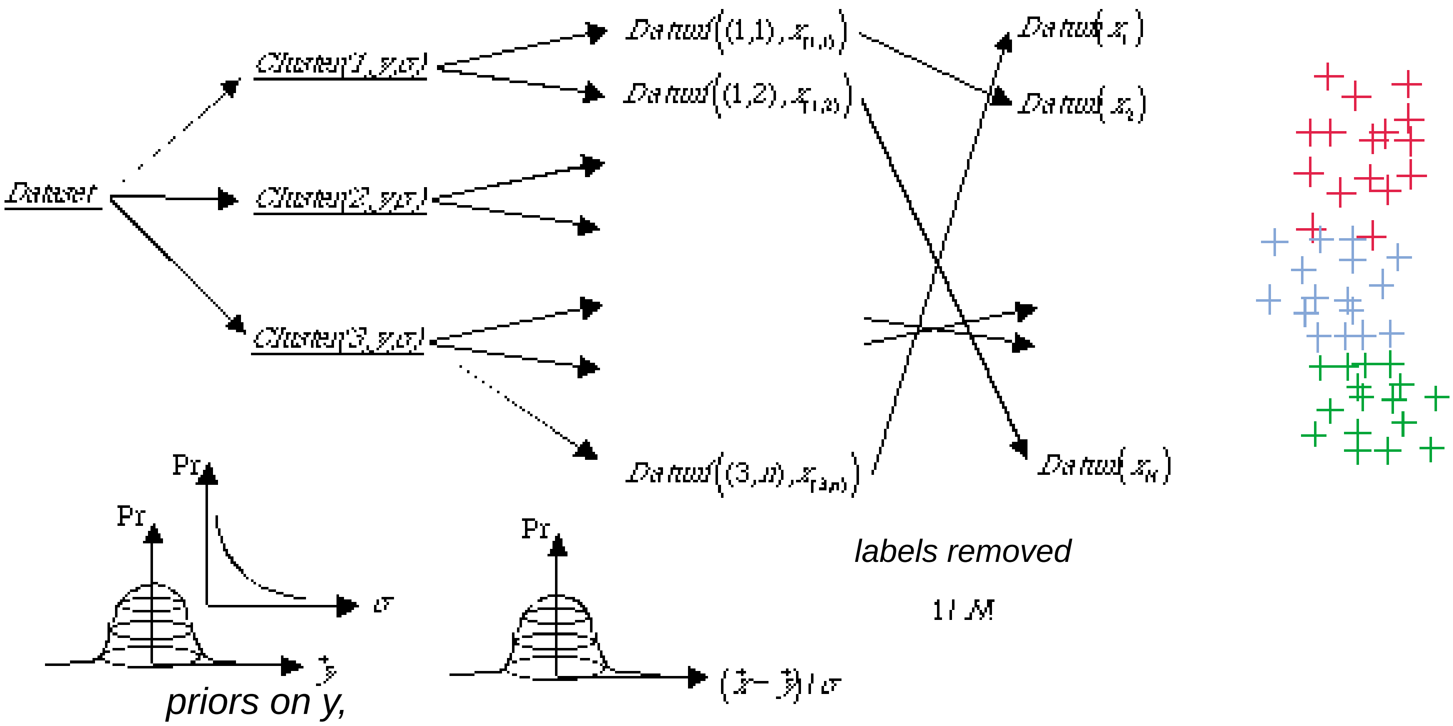
Model portability and re-use (including as subsystem)

Gene Expression Array Modeling

Gene activity levels (here in yeast) are monitored through time
For each gene, a roughly 70-dimensional feature vector arises
Genes are compared by these activity patterns
(green: more expression; red: less expression. Data courtesy Stuart Kim, Stanford)



Genes are clustered hierarchically via a stochastic grammar
(the grammar has a Bayes network analog)



Fitted models represent potential genetic evolution patterns
Models are encoded in XML/Pleodata for inspection by biologists

VIEWPOINT

Aspects of the problem

Primary: Develop the proper language
Declarative, not procedural
Based in neutral mathematical constructs

Develop means of interchange (intermediate formats for editing, display, archiving, and computation)

Develop computational engines

Guiding formalisms

Bayes networks, of course

Parameterized stochastic grammars

Energy minimization

Related work: BUGS

Comes close to addressing these issues
We need a library
Prefer a purely declarative language

Related work: JavaBayes/XML-BIF

Applications require continuous variables
Require continuous functional transformations

MODEL SPECIFICATION (I)

These decompositions have a natural parallel in Bayes net or stochastic grammar formalism

Model list of labeled variables

Text label $\langle L\rangle$ is a means of external or internal reference

$\langle M\rangle \rightarrow \langle L\rangle = \langle V\rangle; \langle L\rangle = \langle V\rangle; ... \langle L\rangle = \langle V\rangle$

Variable Constant, distribution or transformation

$\langle V\rangle \rightarrow \langle C\rangle | \langle D\rangle | \langle T\rangle$

Also, noncircular reference to a labeled variable

$\langle V\rangle \rightarrow \langle L\rangle$

Distribution Familiar families parameterized by variables

$\langle D\rangle \rightarrow \mathrm{Normal}(\langle V\rangle, \langle V\rangle)$

$\langle D\rangle \rightarrow \mathrm{Uniform}(\langle V\rangle, \langle V\rangle)$

$\langle D\rangle \rightarrow \mathrm{Gamma}(\langle V\rangle, \langle V\rangle, \langle V\rangle)$

...

Also

$\langle D\rangle \rightarrow \mathrm{Discrete}(\langle C\rangle, \langle V\rangle, \langle C\rangle, \langle V\rangle, ...)$

the probabilities are constants but the values are variables

Transformation Linear/nonlinear function of a variable

$\langle T\rangle \rightarrow \langle C\rangle * \langle NL\rangle(\langle C\rangle * \langle V\rangle)$

Nonlinearity surrounded by nonsingular linear transforms

The nonlinearity acts coordinatewise:

$\langle NL\rangle \rightarrow [\langle NL1\rangle(\cdot), ... \langle NL1\rangle(\cdot)]$

$\langle NL1\rangle \rightarrow \exp | \log | (\cdot)^p$

EXPRESSIVE POWER

Labels allow construction of DAGs and cyclic graphs.

Aggregation and decomposition of vectors is not allowed.

Random vectors

$$x = \text{Normal}\Big(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}\Big)$$

Composition

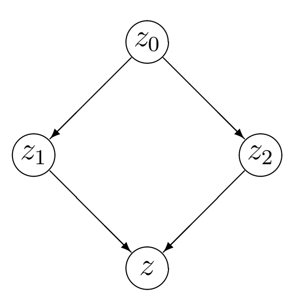
$$y = \exp(\text{Normal}(\text{Normal}(0, 1), 4))$$

Mixtures

$$y = \text{Discrete}(0.1, \text{Normal}(0, 1), 0.9, \text{Normal}(2, 4))$$

DAGs

$$\begin{aligned} z_0 &= \text{Uniform}(0, 1);\\ z_1 &= \text{Normal}(z_0, 1);\\ z_2 &= \exp(\text{Normal}(z_0, 4));\\ z &= \text{Normal}(z_1, z_2); \end{aligned}$$



LANGUAGE SPECIFICATION

Many substrates are possible
support for compositional structure is key

We have chosen XML

Subset of SGML, instantiated for a given application

Looks like another SGML relative, HTML

Content indicated by `<tag> data </tag>` constructs, which may be nested and repeated

Why XML?

- Largely self-explaining document
- Browsers and editors exist (e.g., JUMBO, MSIE5)
  Style sheets allow display in various formats
- Parsers exist in many languages (C, Java, Python, etc.)
- Support for external resources (e.g., data)
- Evolving support for mathematical expressions (MathML)

Encoding

Simple translation to XML

DTD reflects breakdown seen above

Probability Model XML Document Type Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE model [

<!-- Model: list of named variables -->
<!ELEMENT model (variable+)>
<!ATTLIST model name ID #REQUIRED>

<!-- Variable: distribution and optional transform -->
<!ELEMENT variable (dist, transform?)>
<!ATTLIST variable name ID #REQUIRED>

<!-- Transform stuff -->
<!ELEMENT transform (lin_trans?, lin_trans?, xform)>

<!ELEMENT lin_trans (slope, offset?)>
<!ELEMENT slope (#PCDATA)>
<!ELEMENT offset (#PCDATA)> <!-- optional affine part -->

<!ELEMENT xform (from_coord, param)>
<!ELEMENT from_coord (#PCDATA)>
<!ELEMENT param (#PCDATA)>    <!-- which nonlinearity -->

<!-- Distribution stuff -->
<!ELEMENT dist (dim,((val,prob)+ | (mean,covar) | ...))>
<!ATTLIST dist type (#PCDATA) #REQUIRED>
<!ELEMENT dim (#PCDATA)>

<!ELEMENT val  ((variable?) | (#PCDATA)> <!-- discrete -->
<!ELEMENT prob (#PCDATA)>

<!ELEMENT mean  (variable | (#PCDATA))> <!-- normal -->
<!ELEMENT covar (variable | (#PCDATA))>
]>
```

(Some optional attributes have been deleted for clarity)

IMPLEMENTATION

Two operators: `Draw` and `Prob`
Both defined only on labeled variables in the model
Operate by recursive invocation on dependent structures

- `Draw` produces a sample of the variable
  (conditioning not allowed)

Works for any DAG

- `Prob` finds the probability of the variable assuming a value
  (density WRT the appropriate reference measure)

Works for trees but not DAGs

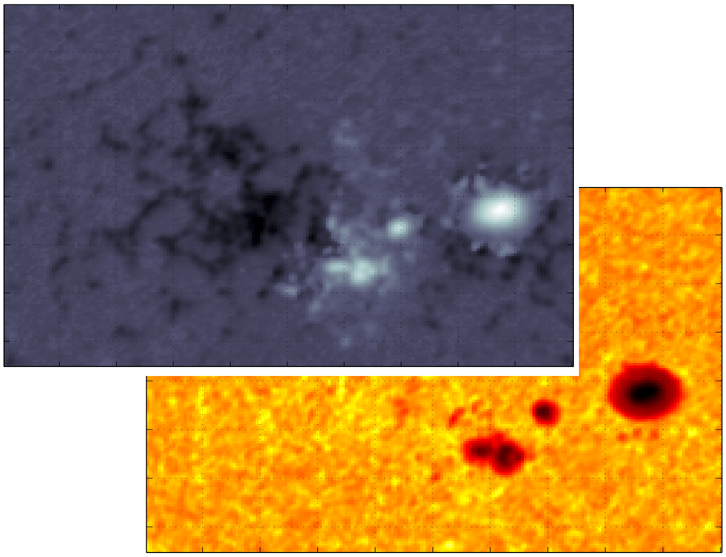
Supports discrete but not continuous integration
  Discrete: needed for finite mixtures
  Continuous: needed to find probabilities like $N(N(0,1),2)$

Summation also done for (finitely) many:one transformations

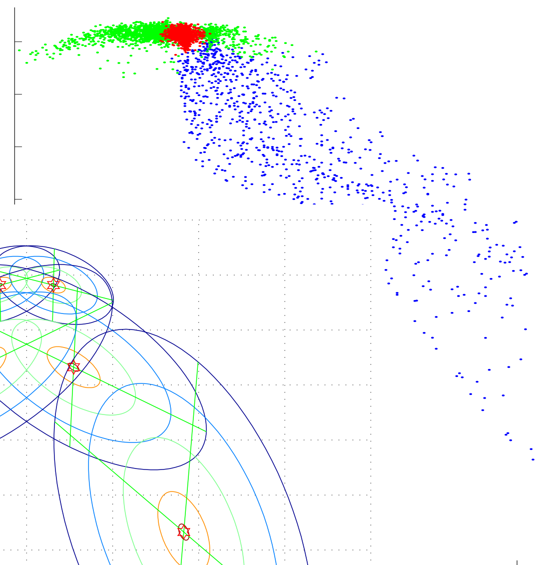
- Environment

Library is in ANSI C
Reads an XML stream with the expat parser

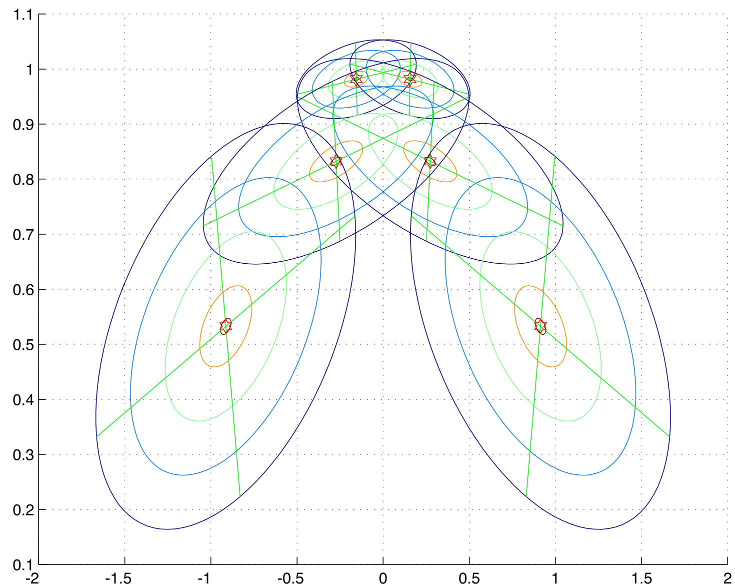
SoHO/MDI Sunspot Identification



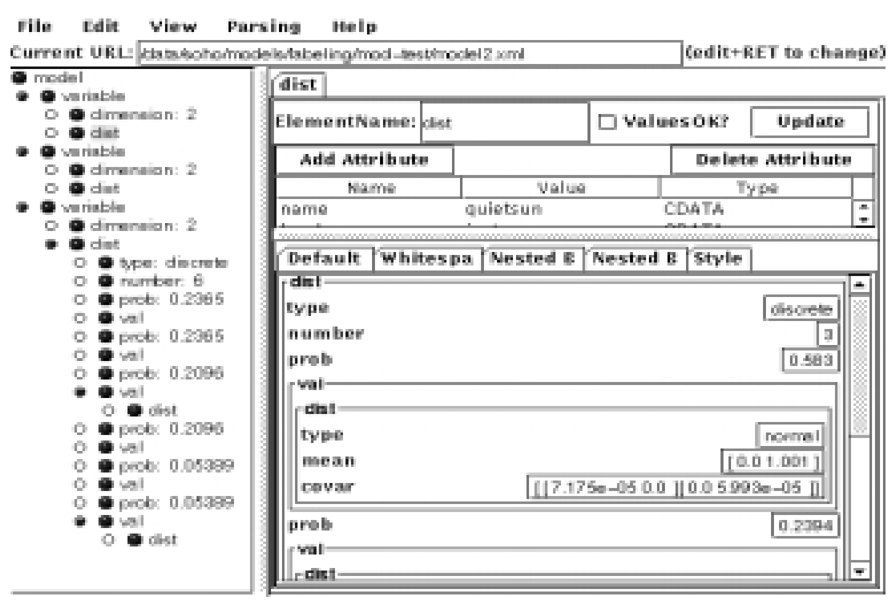
Full-disk images are taken in several modalities by the MDI imager aboard the SoHO satellite
Scientists build models by selecting regions of interest and fitting mixtures to the resulting observables
(red: quiet; green: faculae; blue: spot)



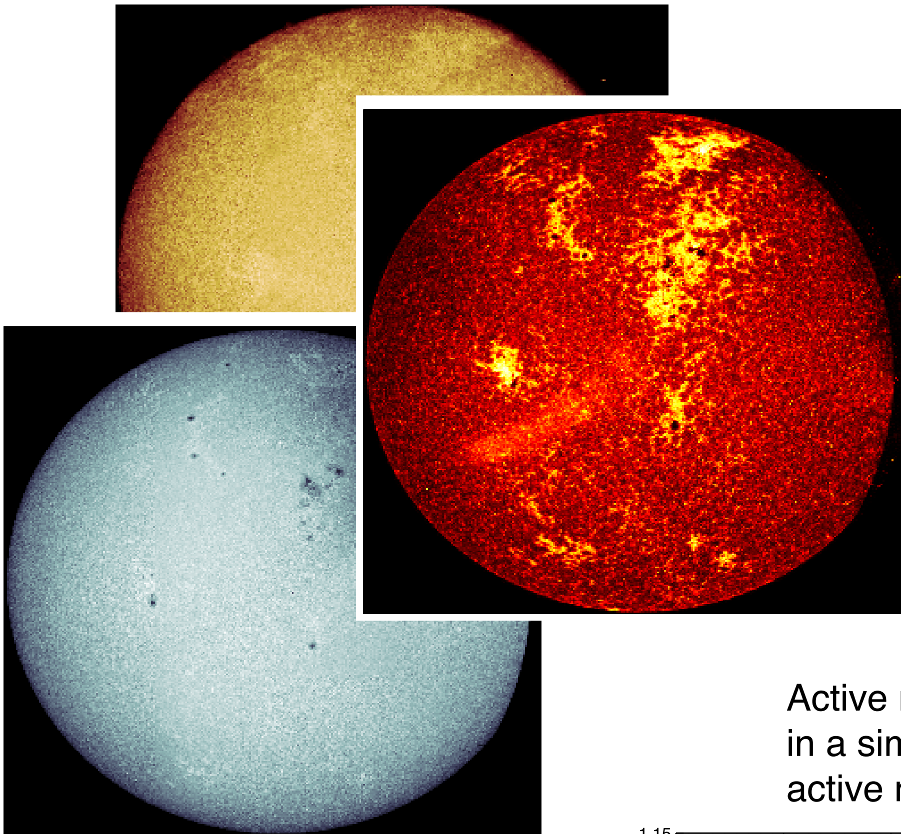
These models, encoded in XML, are the basis for statistically based region identification
Our region-labeling software works by interpreting these model files



Scientists also use model files for documentary purposes and transmission to collaborators

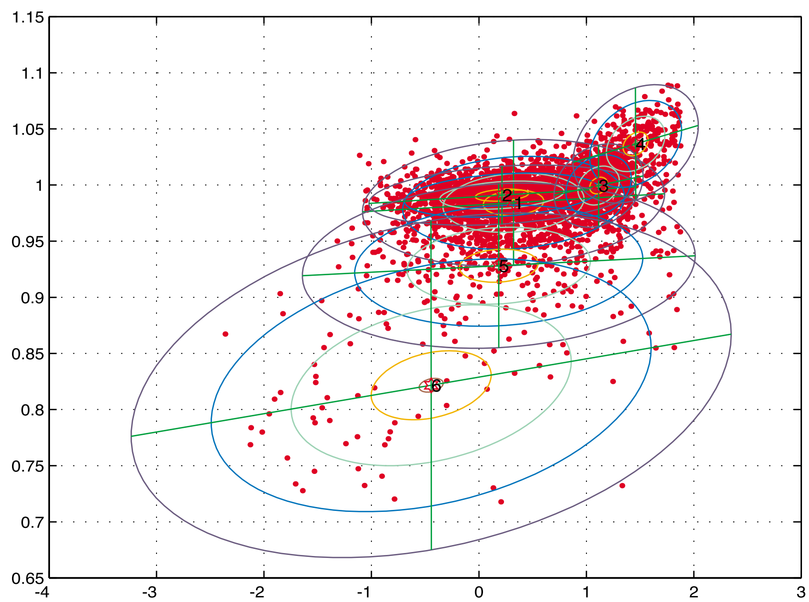


Picard Active Region Identification



As with SoHO/MDI, full-disk images will be taken aboard the CNES Picard satellite

Active region models are determined in a similar way; below is a first-order active region model



Such models are one way to build cross-application analysis mechanisms.

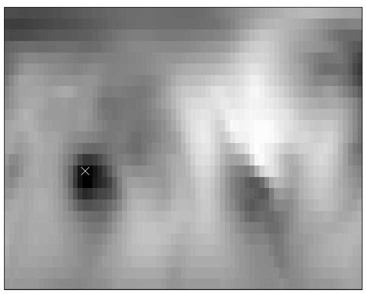
They also allow the analysis community to reach consensus about a given model.

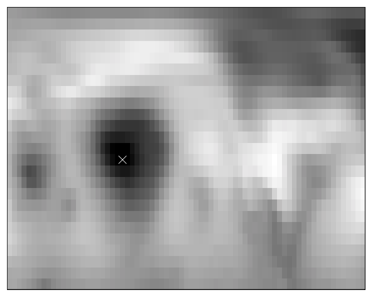
Picard will study temporal changes in solar diameter; region-determination affects the diameter measurement. To maintain the long-term calibration of the Picard measurement, having a definitive region model is essential.

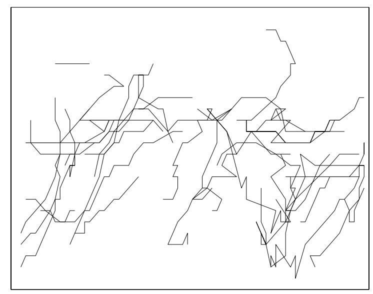
ENVIRONMENTAL TIME SERIES

Object trajectories

Sea-level pressure over the Pacific ($\delta t = 48$ hrs.)
Cyclone center shown by white cross
Right: trajectories from a series of (quantized) observations
Work with Padhraic Smyth, UC Irvine







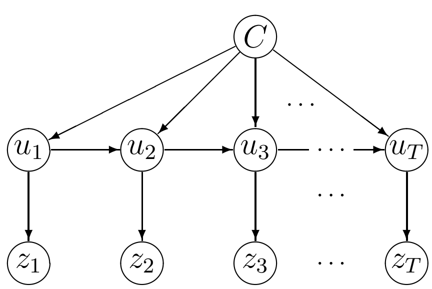
Other examples: sunspot motion, microblock motion from GPS

Modeling trajectories

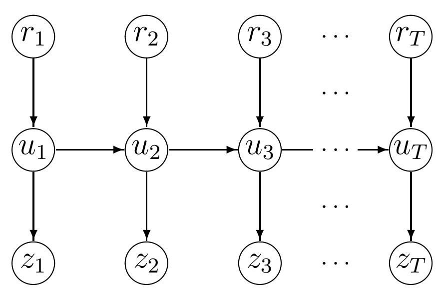
State-based motion models
Include influence of exogenous inputs and observable covariates
Discover motion clusters by uncovering hidden class C

Examples

Generalizations of the Kalman filter as Bayes nets with state u_t



mixed dynamical model



model with exogenous inputs r_t

$\Rightarrow$ need for *spatiotemporal* models

SPATIO-TEMPORAL MODELING (I)

Base concept of random vector is inadequate
Capture concept of variables on structured index sets

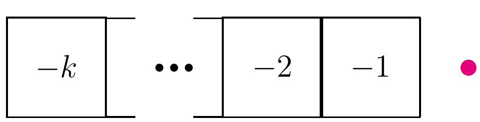
Domain : An index set

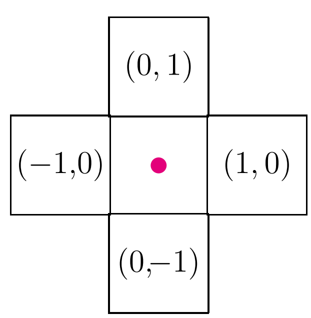
- Principal Examples:
  Any finite set
  Z_n, the first n integers (e.g., time series)
  Z/Z_n, the cyclic version of Z_n
  R, the real numbers

Domains supporting translation play a special role

- Operators on domains give means of combination
  $\cup$, the union
  $\times$, the cross-product
  Allows formation of domains for images, etc.

- *Stencil* is a Domain identifying a local neighborhood
  $\{-k, ..., -2, -1\}$, for a k-order autoregressive model
  $\{(-1, 0), (1, 0), (0, -1), (0, 1)\}$, for a first-order MRF





SPATIO-TEMPORAL MODELING (II)

Field : Mapping on a Domain

Random Field a mapping from a Domain to earlier Variables
...the spatiotemporal generalization of random variable

Principal examples:
Time series are random fields over Z or R
Multispectral images: random fields over $\times(\{1,\ldots k\}, Z_n, Z_n)$
(spectral index does not support translation)

Neighborhood a Field from (Domain, Stencil) to a Domain
...maps $(\text{site}, \text{offset}) \mapsto \text{site}'$, often by translation
...supports *adjacency* for dependence structures

Let M be the neighborhood corresponding to the order-1 MRF
Then $M(i, k)$ is the k-th neighbor of site i
$M(i)$ is the set of all neighbors of site i

unpack operator
...returns the neighborhood M given a Domain and Stencil

MODEL SPECIFICATION

- Simplest models have no conditional dependence:

$$\begin{aligned}&\mathcal{D} = Z_n\\&(\forall i \in \mathcal{D})\, x[i] \sim \text{Normal}(i, 4)\end{aligned}$$

- AR model:

$$\begin{aligned}&\mathcal{D} = Z_n\\&\mathcal{S} = -1\\&M = \text{unpack}(\mathcal{D}, \mathcal{S})\\&(\forall i \in \mathcal{D})\, x[i] \sim \text{Normal}(x[M(i; -1)], 1)\end{aligned}$$

- The standard Potts MRF prior:

$$\begin{aligned}&\mathcal{D} = \times(Z_n, Z_n)\\&\mathcal{S} = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}\\&M = \text{unpack}(\mathcal{D}, \mathcal{S})\\&(\forall i \in \mathcal{D})\, ct[i] = \sum_{k \in M(i)} x[M(i; k)]\\&(\forall i \in \mathcal{D})\, x[i] \sim \text{Discrete}(0, \frac{e^{ct[i]-4}}{e^{ct[i]-4} + e^{-ct[i]}}, 1, \frac{e^{-ct[i]}}{e^{ct[i]-4} + e^{-ct[i]}})\end{aligned}$$

Import just enough mathematical notation to express the models

WISH LIST

Basic Engine

Continuous integration

`Prob` operator for general DAGs using clique-tree algorithm

Usability

Natural editors
e.g., editing stochastic production rules as such

Style sheets for display over WWW

Language

Designing an expressive language is the central question

Declarative language may allow unintended power and features